BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 17/30 — A2

(11) International Publication Number: WO 00/68830

(43) International Publication Date: 16 November 2000 (16.11.00)

(21) International Application Number: PCT/GB00/01532

(22) International Filing Date: 19 April 2000 (19.04.00)

(30) Priority Data:

9910683.3	7 May 1999 (07.05.99)	GB
9910684.1	7 May 1999 (07.05.99)	GB
9910679.1	7 May 1999 (07.05.99)	GB
9910682.5	7 May 1999 (07.05.99)	GB
9910685.8	7 May 1999 (07.05.99)	GB

(71) Applicant *(for all designated States except US)*: ARGO INTERACTIVE LIMITED [GB/GB]; 7 Dukes Court, Chichester, West Sussex PO19 2FX (GB).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: JELBERT, Richard [GB/GB]; 37 Bognor Road, Chichester, West Sussex PO19 2NG (GB).

(74) Agent: ROBINSON, Nigel, Alexander, Julian; D. Young & Co., 21 New Fetter Lane, London EC4A 1DA (GB).

(81) Designated States: JP, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

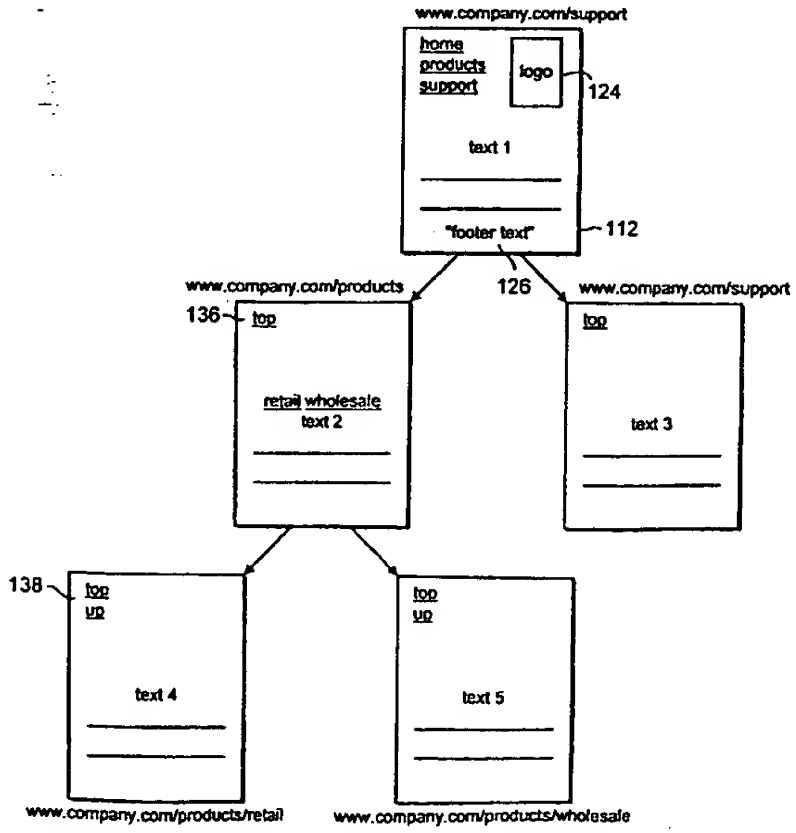
Published

Without international search report and to be republished upon receipt of that report.

(54) Title: MODIFYING A DATA FILE REPRESENTING A DOCUMENT WITHIN A LINKED HIERARCHY OF DOCUMENTS

(57) Abstract

A data processing system for processing documents that include hypertext links (118, 120) to other documents within a hierarchy of documents. A target document being accessed is searched for components (124, 126) that are present in one or more linked documents higher in the hierarchy than the target document. If such repeated components are identified, then they are removed from the target document. Avoiding repeated components reduces the transmission bandwidth and processing requirements as well as the display requirements for the device accessing the document.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

MODIFYING A DATA FILE REPRESENTING A DOCUMENT WITHIN A LINKED HIERARCHY OF DOCUMENTS

This invention relates to data processing systems. More particularly, this invention relates to data processing systems for processing a data file representing a document and including link data items specifying link locations within that document or another document.

It is known to provide a data representation in the form of a linked hierarchy of documents. This is one way in which the page content of the world wide web can be considered. Each page will typically include one or more hypertext links to another page within a hierarchy of documents. At the top of the hierarchy there is typically found the "Home" page. Beneath this Home page there is usually a selection of further pages arranged in accordance with a directory/subdirectory type structure giving information on typically progressively more specific topics. This type of hierarchical arrangement is well understood by users who navigate up and down the hierarchy to find the information in which they are interested.

The overwhelming majority of existing material available for access via the world wide web was created with the intention of being displayed and manipulated using a conventional personal computer. It is desired to make this existing material available for access via different types of device, such as mobile telephones. A problem with this is that the different types of devices may have lower display capabilities and a lower communication bandwidth capability with the result that they struggle to properly display and manipulate the pre-existing material.

Viewed from one aspect the present invention provides a method of processing a data file representing a document, said data file including at least one link data item specifying a linked location within said document or another document, said method comprising the steps of:

(i) accessing a target document within a hierarchy of documents linked by link data items;

(ii) searching one or more linked documents higher in said hierarchy than said target document and linked to said target by a link data item to identify repeated components of said target document that are also present in said one or more linked documents higher in said hierarchy than said target document; and

(iii) removing at least one of said repeated components from said target document.

The invention recognises that it is common for a considerable amount of material within a hierarchy of documents to be repeated. Whilst in conventional computer systems with high resolution displays and large communication bandwidth capabilities, the additional overhead of such repeated material does not pose a significant problem, in less capable systems the repeated material represents a significant processing, bandwidth and display overhead that can degrade performance. The invention addresses this problem by identifying and removing repeated components that are detected as being present higher up in the hierarchy of documents. In this way a user can gain access to a required component by going to the appropriate level in the hierarchy of documents, but the overhead associated with presenting that component at more than one level within the hierarchy can be removed. Users are familiar with traversing a hierarchy of documents to find the information they require and so the removal of the repeated items does not represent a significant usability degradation and is more than outweighed by the increased responsiveness in the way in which the device can access and manipulate the now smaller documents within the hierarchy.

The repeated components could take many different forms. However, the invention is particularly useful in embodiments in which said searching for repeated components comprising searching for one or more of:

repeated link data items;

repeated graphical data items; and

repeated text data items.

It has been found that these types of repeated components frequently occur and may be removed without significantly impacting the usability of the system as a whole.

The hierarchy of documents could be defined in various different ways. However, in one preferred embodiment said documents are internet web pages each having an associated universal resource identifier, said hierarchy of documents following a hierarchy defined by said universal resource identifiers.

Internet web sites are typically structured by the author to follow such a hierarchy based upon the universal resource indicator. Accordingly, removing repeated components based upon this same hierarchy is often found to produce good

results in terms of the intuitive nature of where particular components will first appear within the hierarchy.

As an alternative, said hierarchy of documents follows an order in which said documents are accessed by a user in a session with documents accessed earlier in said session being positioned higher in said hierarchy than documents accessed later in said session.

Such a session based approach to defining the hierarchy is better suited to some patterns of access by users. A session may be thought of as being bounded by a users interaction with the system to perform a certain task. In particular, many users have their own bookmarked pages that they frequently visit. These bookmarked pages may not be at the top of a universal resource indicator hierarchy and yet are the most frequent starting point for that user when moving through that web site. Taking a session based approach to establishing the hierarchy recognises this pattern of usage and makes it more likely that the user will be able to quickly find the component they are looking for even though they may not start from the top of the universal resource identifier hierarchy.

The top of a session based hierarchy can be chosen in various different ways. In preferred embodiments said hierarchy uses as its highest document one of:

a first document accessed in said session;

a first document accessed within a predetermined preceding period within said session: and

a first document accessed within a predetermined number of previously accessed documents within said session.

The above ways of defining the top of a session based hierarchy each have advantages in particular circumstances and a user may be allowed the option to select a particular way in which they wish their session based hierarchy to be defined.

A common repeated component within a hierarchy of documents is a navigation bar that presents buttons representing links to various points within the hierarchy of documents. It is common for the same navigation bar to be presented upon every page of a web site. Navigation bars often include a significant graphical content that imposes a processing and bandwidth load that is disadvantageous for a display device of a more limited capability. Accordingly, it is desirable to remove repeated items within navigation bars and the like. However, in order to preserve the ability of a user to rapidly and intuitively navigate through the site, preferred

embodiments serve to add one or more of an up link to a document one higher within a hierarchy and a home link to the document highest within the hierarchy.

The processing and bandwidth overhead associated with adding one or both of these links to the pages is more than outweighed by the benefit in usability achieved.

Whilst the link data items could take a variety of forms, it will be appreciated that the invention is particularly well suited to embodiments in which the link data items are hypertext links. Similarly, whilst the invention could be used on a stand alone system, it is particularly useful in embodiments in which the data file is retrieved from a source computer server via a computer network. In such embodiments a proxy server disposed within the computer network between the source computer server and a client computer requesting the data file is often able to provide the processing and storage capability to perform the steps of accessing, searching and removing without placing a significant extra burden upon the client computer itself. However, as the processing capabilities of client computer devices, such as mobile wireless devices, improve, it becomes increasingly possible that the steps of accessing, searching and removing could be performed on the client computer itself. The client computer may have different capabilities from that for which the document was originally intended or the document may be display independent.

Viewed from another aspect the present invention provides an apparatus for processing a data file representing a document, said data file including at least one link data item specifying a linked location within said document or another document, said apparatus comprising processing logic for performing the steps of:

(i) accessing a target document within a hierarchy of documents linked by link data items;

(ii) searching one or more linked documents higher in said hierarchy than said target document and linked to said target by a link data item to identify repeated components of said target document that are also present in said one or more linked documents higher in said hierarchy than said target document; and

(iii) removing at least one of said repeated components from said target document.

An embodiment of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 schematically illustrates a computer network:

Figure 2 schematically illustrates a system for adding categorising data to a data file representing a document;

Figure 3 illustrates a link data item and associated keywords;

Figure 4 schematically illustrates a hierarchical category database;

Figure 5 illustrates a category data entry;

Figure 6 illustrates how a web page may be modified using category data to filter out links known to be unwanted of less wanted by a user;

Figure 7 is a flow diagram illustrating the addition of category data to a document;

Figure 8 schematically illustrates a system for adding output graphical data to a document;

Figure 9 illustrates a low resolution display device showing a document before and after addition of icons in accordance with category data;

Figure 10 is a flow diagram illustrating the addition of output graphical data items in association with link data within a document;

Figure 11 schematically illustrates modifying display text associated with a link data item into a more readable form;

Figure 12 shows a flow diagram illustrating the process of modifying display text into a more readable form;

Figure 13 illustrates various examples of text modifications that may be performed;

Figure 14 illustrates an unmodified hierarchy of documents including repeated components;

Figure 15 illustrates a modified form of the hierarchy of Figure 14 in which repeated components have been removed;

Figure 16 illustrates the comparison between a universal resource identifier based hierarchy and a session based hierarchy;

Figure 17 is a flow diagram showing the process for removing repeated components within a hierarchy; and

Figure 18 schematically illustrates a data processing apparatus that may serve as a client computer.

Figure 1 illustrates a computer network 2. This computer network 2 may be a portion of the internet in which internet web pages in the form of HTML data files are transmitted between source servers 4 and client computers 6, 8. A proxy server 10 is

disposed between the source servers 4 and the client computers 6, 8. The client computer may be a normal desktop computer 6 for which the internet web pages are primarily designed and intended. The client computer may also be in the form of an internet-enabled mobile telephone 8 connected via a radio link 12 to the computer network 2.

The mobile phone 8 connects via the proxy server 10, and the proxy server 10 may detect (e.g. via user id and password details) that the link from the mobile phone 8 as a client computer is to a device having a smaller and less capable display than a full desktop computer 6. Accordingly, the proxy server 10 is able to perform additional processing steps on the internet web pages fetched from the source servers 4 before they are passed to the mobile telephone 8 so that they can be adapted to be more usefully displayed on the mobile telephone 8. It will be appreciated that if the processing capabilities of the mobile telephone 8 were greater and the radio bandwidth sufficient, then the full internet web pages could be transmitted to the mobile telephone 8, which may then conduct its own processing of those pages to put them into a form more suitable for display on its smaller display output.

Figure 2 schematically illustrates how a data file representing a source document 14 may be processed by a link categoriser 16 to generate an output document 18 that has category data added to it. It will be appreciated that the link categoriser 16 will typically take the form of a general purpose computer executing software written to perform the function of adding the category data to the documents. The link categoriser 16 uses a category-to-keyword database 20 which enables keywords identified within the source document 14 to be mapped to appropriate categories. The category-to-keyword database 20 can be in the form of a hierarchical database with each category data entry having the keywords associated with that category data entry related thereto and with score values for each associated keyword. The link categoriser 16 also uses a user-to-category database 22 which enables the link categoriser to perform other functions, such as modifying the source document in a way that removes or adds data known to be of particular interest the user concerned.

Figure 3 illustrates a link data item 24 that is typically embedded within a HTML document. The link data item 24 includes a universal resource identifier 26 and display text 28. If display text 28 is present, then this is what will be displayed as the hypertext link in the document. If display text 28 is not present, then the universal resource identifier 26 will be displayed.

The keywords within the link data item 24 are identified by processing the link data item 24 by removing all punctuation and replacing this with spaces. The resulting stream of keywords 30 can then be input to the keyword-to-category matching database 20. The category-to-keyword database 20 can be arranged as a relational database making the analysis of the keywords sufficiently rapid to be performed in real time by the proxy server 10.

Figure 4 schematically illustrates the hierarchical nature of the category database 20. In particular, a category such as "Transport" can be broken down into a number of sub-categories such as "Car", "Motorcycle", "Bicycle", "Lorry", and "Van". Each of these sub-categories can be further broken down as illustrated. The hierarchy could have a varying depth depending upon the required degree of specificity traded off against the processing and data storage requirements as well as the likelihood of a highly specific categorisation in fact being correct.

Figure 5 schematically illustrates a particular category data entry within the category-to-keyword database 20. In this case, the category data 32 is associated with a sequence of keywords 34 each having an associated score value 36. The keywords 30 with the link data item 24 are matched against the keywords 34 and the score values 36 for each match of a category data entry 32 added together. The category data entry 32 having the highest score is deemed to be the match.

Returning to Figure 2, when the category data entry 32 that produces the best match has been identified, then category data 38 in the form of a metatag is inserted into the document 18 in association with the link data item 24 that has been analysed. The category data 18 thus gives a representation of the subject matter to which the link data item 24 relates. This information is highly useful to other processes performed by the proxy server 10. In particular, the proxy server 10 might automatically insert a graphical item before each hypertext link to assist in faster recognition of links of interest. The proxy server 10 could filter out categories that are known to be unsuitable or undesired for the user, for example if the reader is known within the user-to-category database 22 to not want information concerning cars. The proxy server 10 can also record information regarding the categories of links followed by a user while viewing hypertext documents and so assemble a profile of the user's interest such that other material of possible interest to the user, such as targeted advertising, may be presented to the user. Another use that can be made of such user profiling information is pre-fetching of information relevant to the user's

interests. Using pre-fetching, the proxy server 10 may automatically collect and store information that the user is likely to want to view before they request it. If they do then request this information, it can be delivered more quickly. If they do not request the information, then the information can be discarded.

Figure 6 shows how an original web page 80 containing ten hypertext links can be modified into a page 82 more suited to display using a smaller display window 84 by the removal of hypertext links detected as either not wanted or less likely to be wanted by a user. This is done by comparing the category data 38 associated with each link with the user preference data stored in the user to category database 22. The user to category database 22 can contain preference data obtained by the user specifying categories of link in which they are not interested and do not wish to display. Alternatively or additionally, the user to category database 22 can be automatically built up by the proxy server 10 keeping a record of the categories of the links that a user follows, e.g. by dynamically user profiling the categories of interest. Thus, categories stated or observed to be of little interest to a user can be removed from the page 82 so making better use of the limited bandwidth and display resources. This sort of content filtering may also be used to block material, such as by a parent wishing to prevent access to unsuitable material by a child.

Figure 7 is a flow diagram illustrating the process of adding category data to a source document. At step 52, the source document is fetched via the network link from the source server 4. The proxy server 10 at step 54 processes the source document to identify the link data items 24 within it and isolate the keyword data within those link data items 24. At steps 56 and 58, the proxy server applies a series of rules to the keywords identified within the link data item 24 to determine whether they are sufficiently specific to enable a proper categorisation to be made. An example of the rules applied are as follows:

1) Initially everything is neat, i.e. is initialized in a state termed "neat";

2) It is ruled as being not neat it the length of the text is greater than 10 AND the length to space ratio is greater than 10:1;

3) It is ruled as being neat if the text is "entertainment";

4) It is ruled as being not neat if the text is "image" followed by a number;

5) It is ruled as being not neat if the length of the text is less than 4 characters;

6) It is ruled as being not neat if the number of underscores exceeds the number of spaces;

7) It is ruled as being not neat if the text beings with "http://";

8) It is ruled as being not neat if the text is enclosed with quotes;

9) It is ruled as being not neat if the text beings with "image map ";

10) It is ruled as being not neat if the text is "default".

In addition, there are additional rules that may be added for specific geographical locations, e.g:

11) It is ruled as neat if the text contains "Island";

12) It is ruled as neat if the text contains "Kanagawa-Ken".

Both of these (and also some of the specific rules) may be added in a category such as 'rules specific to sites'.

If sufficient information is present, then processing proceeds to step 60. If sufficient information is not present, then the proxy server 10 fetches the title data of the target location identified by the link data item 24 to derive additional keywords from that title data. The entire document indicated by the link data item need not be fetched. This contrasts to spidering in which the entire document pointed to by a link data item is fetched and analysed.

At step 60, the proxy server/link categoriser 16 looks up the keywords identified within the category-to-keyword database 20 and scores each possible category. At step 62, the category with the highest score is selected to be associated with the link data item 24. At step 64, a metadata tag identifying the category selected at step 62 is inserted into the document in association with the link data item 24.

Figure 8 schematically illustrates a system for modifying the graphical data contents of a document. A source document 40 is accessed from a source server 4 via an internet link. The source document 40 is in the form of a HTML document representing an internet web page. The source document 40 may contain GIF files, JPEG files and bitmap files as part of its source graphical data content. The source document 40 includes category data 38 classifying the link data items 24 as added by the processing discussed above.

A graphical icon allocator 42 receives the source document 40 and removes all or some of the source graphical data items. The graphical icon allocator 42 then accesses a category-to-icon database 44 where icons suitable for association with each

link data item 24 within the source document 40 are identified using the category data 38 embedded within the source document 40. When an output graphical data item has been identified from the category-to-icon database 44, then data identifying this icon 46 is inserted as a metatag into the output document 48. The data identifying the output graphical data item 46 may be merely an identifier for an icon which is built into the known display device 8, or alternatively it may be data giving sufficient information to specify the appearance of the icon without this already being embedded within the display device 8.

It will be appreciated that the graphical icon allocator 42 will typically take the form of software operating on a general purpose computer, such as the proxy server 10. If the processing capabilities of the client computer 8 are sufficient and sufficient bandwidth is available, then the source document 40 may be transmitted to the client computer 8 in its entirety and the processing illustrated in Figure 6 performed wholly within the client computer 8.

Figure 9 illustrates a small low resolution display device 50, such as the small LCD display of a mobile telephone 8. The left hand portion of Figure 7 illustrates a text-only web page showing a series of hypertext links with all of the graphical data from the source page removed. The usability of such a display is poor compared to the original source document 40 as users derive considerable information from the graphical data content of a page.

Using the present invention, the links within the page can be categorised and then appropriate icons associated with each link. These icons can be built into the mobile telephone 8 itself such that they do not need to be transmitted to the client computer in their entirety. A code identifying a particular built-in icon can merely be added as the data 46 in the output document 48.

Figure 10 is a flow diagram illustrating the processing of graphical data items. At step 66, the proxy server 10 fetches a source document 40. At step 68, the proxy server/graphical icon allocator 42 removes all non-text data from the source document 40. At step 70, the graphical icon allocator maps the category data 38 to icons to be associated with the link data item 24 using the category-to-icon database 44. At step 72, the icon identifying data is inserted as a metatag 46 within the output document 48. At step 74, the resulting output document 48 including text data and associated icon data is transmitted to the client computer 8. At step 76, the client computer 8

processes the received document and displays the text with its associated icons next to the link data items. The icons can be built-in icons within the client computer 8 itself.

Figure 11 illustrates a source document 78 in the form of an internet web page intended by the author to be displayed and manipulated using a conventional personal computer. Within the document 78 there is a link data item 80 in the form of a hypertext link to a large image file. A small thumbnail representation 82 of the full image file is also shown. When a user accesses this web page 78 on a conventional personal computer, then the thumbnail representation 82 in combination with the display text of the link 80 gives sufficient information for the user to understand the link being made. However, if the web page 78 is modified to produce a modified page 84 in which graphical data has been removed, then the initial display text 86 associated with the link 80 may not be sufficient to enable a user to properly understand the connection being made.

The system identifies the links within the web page 78 and performs tests upon the initial display text associated with each link to determine characteristics indicative of insufficient readability. In the case of the initial display text 86 shown in Figure 11, then this may fail the test of comprising too many characters within a word or of including a capital letter following a lower case letter within the middle of a word. The initial display text 86 having been identified as not sufficiently readable, the title 88 of the page to which the link relates is accessed and this title used as further text in place of the initial display text 86. The title 88 is itself subject to an assessment of its readability and only if it passes this determination does it remain as a replacement for the initial display text 86. If the further text 88 fails the readability test, then the initial display text is reverted to for the link 80.

The above technique uses a system of computer software through which users are required to fetch hypertext documents that they wish to read. Typically this is in the form of an intermediate "proxy server", but a stand-alone mode of operation can also be envisaged. The system processes the hypertext pages as they are transferred from the storage location to the reader. After identifying the links in the hypertext document, the textual part of the hypertext link (i.e. the text which the user would select in order to go to the linked document) is checked to see if it is readable. This can be done in a number of ways, including (but not limited to):

- the number of underscores is greater than the number of spaces;

- the text is less than a certain number of characters long;
- the text is longer than a certain number of characters long;
- the average number of characters per word is greater than a certain limit;
- the text contains words which have capital letters after lowercase letters in the same word (e.g. gooSE);
- the text contains words which are not in a dictionary;

A combination of the above rules can be used to score the link in terms of readability, and if the score is above a threshold, then an alternative to the text is sought. This can also be done in several ways, including (but not limited to):

- fetching the linked hypertext document and retrieving the document's title (should one exist), or the first line of the text in the document;
- substituting the text with different text from a dictionary (stored in a file coupled to the proxy server e.g. a keyword to further text mapping);
- replacing with the title of the current document (should one exist);
- using a filename with its file type suffix removed.

If the further text that is to replace the initial display text is deemed more unreadable then the initial display text, then the initial display text is kept in place, and either no substitution takes place, or an alternative substitution is used.

Figure 12 shows a flow diagram illustrating the technique of improving the readability of the display text associated with links.

At step 90 a page to be accessed is fetched from a remote computer server. At step 92 the fetched page is searched to detect link data items (hypertext links) and the initial display text associated with these links is determined. At step 94 the readability rules described above are applied to the initial display text of each link. At step 96 a determination is made as to whether or not the initial displayed text passes the readability rules. If the initial display text does pass the readability rules, then the process proceeds to step 98 where the output page is generated.

If the initial display text does not pass the readability rules at step 96, then step 100 is used to replace the text with further text derived in dependence upon the link item data, such as by using the replacements described above. These candidate replacements can be applied in turn with each candidate replacement being tested by steps 102 and 104 to determine whether or not it passes the readability test. If it does

pass the readability test at step 104, then the replacement candidate is used as the further text to replace the initial display text within the link data item and an output page including this further text is produced at step 98. If the candidate replacement text does not pass the readability text, then the next candidate replacement text will be tried providing step 106 does not determine that all the candidates have been exhausted. If step 106 does determine that all the candidate replacement text have been exhausted, then step 108 reverts to the initial display text and the output page is produced using this initial display text at step 98.

Figure 13 schematically illustrates how some initial display text may be modified into forms more readily readable. In example A, a file name containing a mixture of numbers and underscore characters and exceeding a predetermined length is replaced by the title of the page to which it points. In example B, an initial display text that is too short to be useful is replaced with category data associated with the link and derived as described above. In example C, an initial display text that is too long to be usefully displayed on a mobile telephone is replaced by a text that uses keywords selected from the initial longer text. Finally, in example D, a file name is replaced by the file name minus its file type suffix.

As previously described, it will be appreciated that the processing described above to improve the readability of the display text associated with a link data item may be performed either on a proxy server using the superior processing and storage capabilities of that proxy server, or upon the client device itself. As the client devices improve in their capability, it will be natural for more processing to take place upon the client device and so remove the need for the connection to have to be made through a particular proxy server.

Figure 14 schematically illustrates an internet web site in the form of a hierarchy of documents. Each page has an associated universal resource identifier 110 with a form similar to a directory/subdirectory structure. The hierarchy illustrated starts with a company home page 112 and progresses to a products page 114 and a support page 116 via respective hypertext links 118 and 120. The hypertext links 118 and 120 together with a home page link 122 form a navigation bar that appears on all of the pages of the web site. A company logo 124 and a standard footer text 126 also appear on all pages of the web site.

The product page 114 includes two further hypertext links 128 and 130 that respectively point to pages 132 and 134 giving details of retail and wholesale

products. Each of the pages 112, 114, 116, 132 and 134 also includes its own unique text.

It will be appreciated that when processing and bandwidth resources as well as display device resources are limited, then the repeated transmission, processing and display of items such as the company logo 124 and the footer text 126 represent a significant overhead. Assuming that a user enters the site at page 112, then they are initially presented with the opportunity to progress to the support page. If instead the user progresses to the products page 114, then it is reasonable to assume that they are not interested in support. Accordingly, it is wasteful to display the link 120 to the support page 116 on the product page 114 as well as on the home page 112.

Figure 15 illustrates the web site shown in Figure 14 but this time modified such that repeated components lower down in the hierarchy are removed, i.e. in this arrangement components appear upon their first occurrence when moving down the hierarchy but are thereafter removed. As an example, the company logo 124 appears on the home page 112, but does not appear on any of the pages lower in the hierarchy. Similarly the footer text 126 appears only on the home page 112 and has been removed from the lower pages. The links 118, 120 and 122 that form the navigation bar appear only on the home page 112. On the lower pages, a link 136 is added linking to the top page in the hierarchy. Where there is a page above the current page that is not the top page, then an uplink 138 is also added.

It will be seen from Figure 15 that the content of the pages below the home page 112 has been significantly reduced so enabling them to be more rapidly transmitted to a client computer and conveniently and rapidly manipulated on that client computer. Nevertheless, all of the content of the original web site illustrated in Figure 14 is present within the modified web site shown on Figure 15 at some point within that web site.

Figure 16 schematically illustrates how a web site may be placed into a hierarchy based upon the universal resource indicators as compared to a session hierarchy. On the left hand side of Figure 16 is shown a hierarchy derived from the universal resource identifiers. The letters next to each node indicate a unique page. The vertical position within the illustrated hierarchy denotes the position within the hierarchy. The numbers next to each node represent the order in which the pages are accessed during a user session. With the hierarchy based upon the universal resource identifier, page a is at the top of the hierarchy and page e is towards the centre.

Compared to the universal resource identifier hierarchy, the session hierarchy illustrated in the right hand portion of Figure 16 shows a hierarchy in which the first pages to be accessed are disposed higher within the hierarchy. Accordingly, since the first page accessed (e.g. through a bookmark) was page e, this is at the top of the hierarchy. A user may subsequently traverse the entire web site in the order shown by the numbers. The pages are arranged in the session hierarchy according to these numbers with pages at the same horizontal level indicating the same position within the hierarchy.

Hypertext documents are viewed in some sequence by each reader, moving from one to another by choosing "links" within each page. Where some information is presented on an early page and then ignored by the reader, it is reasonable to assume that they are not interested in it. Also, many modern hypertext document systems (sometimes called "web sites") are designed in a hierarchical form. There may be pages to list the sections of the web site, and more to list each sub-section, followed by pages containing actual content. Either such a hierarchy or the historical tracking of a user's reading can be employed to assist the system predicting which pages a reader should already have read, if historical tracking information has not been recorded for them.

The present technique uses a system of computer software, through which users are required to fetch hypertext documents that they wish to read. Typically this is in the form of an intermediate "proxy server", but a stand-alone mode of operation can also be envisaged. The system processes the hypertext pages as they are transferred from the storage location to the reader, removing parts, recording what it has found, and performing other tasks.

Once a hypertext document has been requested by the user and subsequently reviewed by the system, the system examines the hierarchy in which the page exists on the basis of the document's Uniform Resource Identifier (URI). This URI, or some similar information appropriate to the hypertext system being used, should uniquely identify the page and provide some information about the hierarchy in which it exists. The system fetches each page that is above the requested one in the hierarchy (sometimes called "parent" pages), and makes a note of discrete units of information on each page. It may only note links to other pages, but divisions of other information such as images and/or footnotes are also envisaged. If the reader's

activity is being recorded, then pages they have already viewed may be considered instead of parent pages of the current document.

Once a note has been made of the information units on each page, those units that are present on parent pages are removed from the one requested by the reader. One or more new links are added to the current page to ensure that the reader has the opportunity to return to pages which do contain the links, should they wish to use them.

The advantage of this a procedure is that each document will be reduced to a more manageable size without removing significant information from it, and without requiring special preparation by the hypertext author. This is important for small devices that are technically limited and very different from the majority of readers for whom such authors write.

If the system is configured to work with a historical record of pages viewed by the reader, the oldest page considered as part of the link removal may either be the first page seen, the first seen within a certain time, e.g. ten minutes, or the N'th last page, perhaps the tenth last. It would not consider any page viewed after the first viewed of the current page (nor of course would it treat the current page as a previous one). This ensures that if the user goes "back" to a previous page, they will not lose all of the links on it.

Figure 17 is a flow diagram illustrating the above process. At step 140 a target document is accessed. At step 142 the components making up that target document are compared with components known to be in document higher in the hierarchy than the target document. The contents of the components higher in the hierarchy may be determined by fetching those pages in dependence upon their universal resource identifier if they have not already been so fetched or may be determined on a user session basis as previously described.

At step 144 items within the target document found to be repeated components that are present in documents higher in the hierarchy are removed. At step 146 hypertext links to the top of the hierarchy and possibly also to one step up in the hierarchy are added. At step 148 the output page is generated.

Figure 18 schematically illustrates a client data processing apparatus, such as a mobile telephone. The client device 150 will typically include a central processing unit 152, a read only memory 154, a random access memory 156, a display driver 158, a display 160, a communications interface 160 and an antenna 162. The central

processing unit 152, the read only memory 154, the random access memory 156, the display driver 158 and the communications interface 160 are connected via a common bus 164. The read only memory 154 may form a computer program storage device holding a computer program for controlling the central processing unit 152 to carry out the processing described above where the processing is client based. The random access memory 156 will be used as working storage. The display 160 may be of a reduced size and resolution compared to a typical personal computer, e.g. it may be a low resolution LCD screen as typically found on present day mobile telephones, or just a small display per se. The communications interface 160 illustrated is a wireless interface that is linked to the proxy server 10 via the antenna 162.

CLAIMS

1. A method of processing a data file representing a document, said data file including at least one link data item specifying a linked location within said document or another document, said method comprising the steps of:

(i) accessing a target document within a hierarchy of documents linked by link data items;

(ii) searching one or more linked documents higher in said hierarchy than said target document and linked to said target by a link data item to identify repeated components of said target document that are also present in said one or more linked documents higher in said hierarchy than said target document; and

(iii) removing at least one of said repeated components from said target document.

2. A method as claimed in claim 1, wherein said searching for repeated components comprising searching for one or more of:

repeated link data items;

repeated graphical data items; and

repeated text data items.

3. A method as claimed in any one of claims 1 and 2, wherein said documents are internet web pages each having an associated universal resource identifier, said hierarchy of documents following a hierarchy defined by said universal resource identifiers.

4. A method as claimed in any one of claims 1 and 2, wherein said hierarchy of documents follows an order in which said documents are accessed by a user in a session with documents accessed earlier in said session being positioned higher in said hierarchy than documents accessed later in said session.

5. A method as claimed in claim 4, wherein said hierarchy uses as its highest document one of:

a first document accessed in said session;

a first document accessed within a predetermined preceding period within said session; and

a first document accessed within a predetermined number of previously accessed documents within said session.

6. A method as claimed in any one of the preceding claims, further comprising the step of adding an up link data item to said target document, said up link data items specifying a linked location within a document one higher within said hierarchy.

7. A method as claimed in any one of the preceding claims, further comprising the step of adding a home link data item to said target document, said home link data items specifying a linked location within a document highest within said hierarchy.

8. A method as claimed in any one of the preceding claims, wherein said link data items are hypertext links.

9. A method as claimed in any one of the preceding claims, wherein said data file is retrieved from a source computer server via a computer network.

10. A method as claimed in claim 9, wherein said steps of accessing, searching and removing are performed by a proxy server disposed within said computer network between said source computer server and a client computer requesting said data file.

11. A method as claimed in claim 9, wherein said steps of accessing, searching and removing are performed by a client computer which requests said data file from said source computer server.

12. A method as claimed in any one of claims 10 and 11, wherein said client computer has a user display with different display capabilities than those of a display for which said document is intended or said document is display independent.

13. A method as claimed in claim 12, wherein said client computer is a wireless device.

14. Apparatus for processing a data file representing a document, said data file including at least one link data item specifying a linked location within said document or another document, said apparatus comprising processing logic for performing the steps of:

(i) accessing a target document within a hierarchy of documents linked by link data items;

(ii) searching one or more linked documents higher in said hierarchy than said target document and linked to said target by a link data item to identify repeated components of said target document that are also present in said one or more linked documents higher in said hierarchy than said target document; and

(iii) removing at least one of said repeated components from said target document.

15. Apparatus as claimed in claim 14, wherein said data file is retrieved from a source computer server via a computer network.

16. Apparatus as claimed in claim 15, wherein said processing logic is part of a proxy server disposed within said computer network between said source computer server and a client computer requesting said data file.

17. Apparatus as claimed in claim 15, wherein said processing logic is part of a client computer which requests said data file from said source computer server.

18. A computer program storage medium storing a computer program for controlling a data processing apparatus to perform the method as claimed in any one of claims 1 to 13.

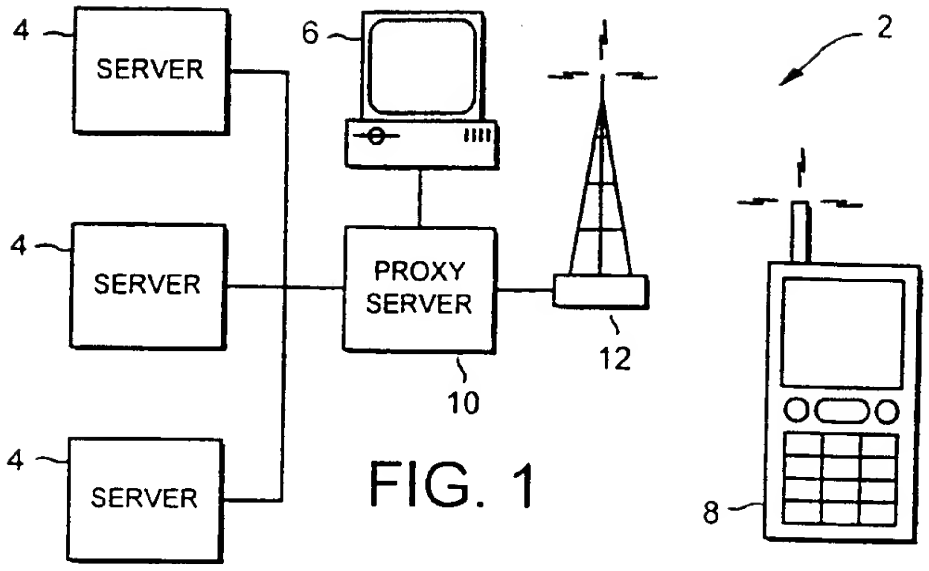

4
SERVER
4
SERVER
4
SERVER
6
PROXY SERVER
10
12
2
8

FIG. 1

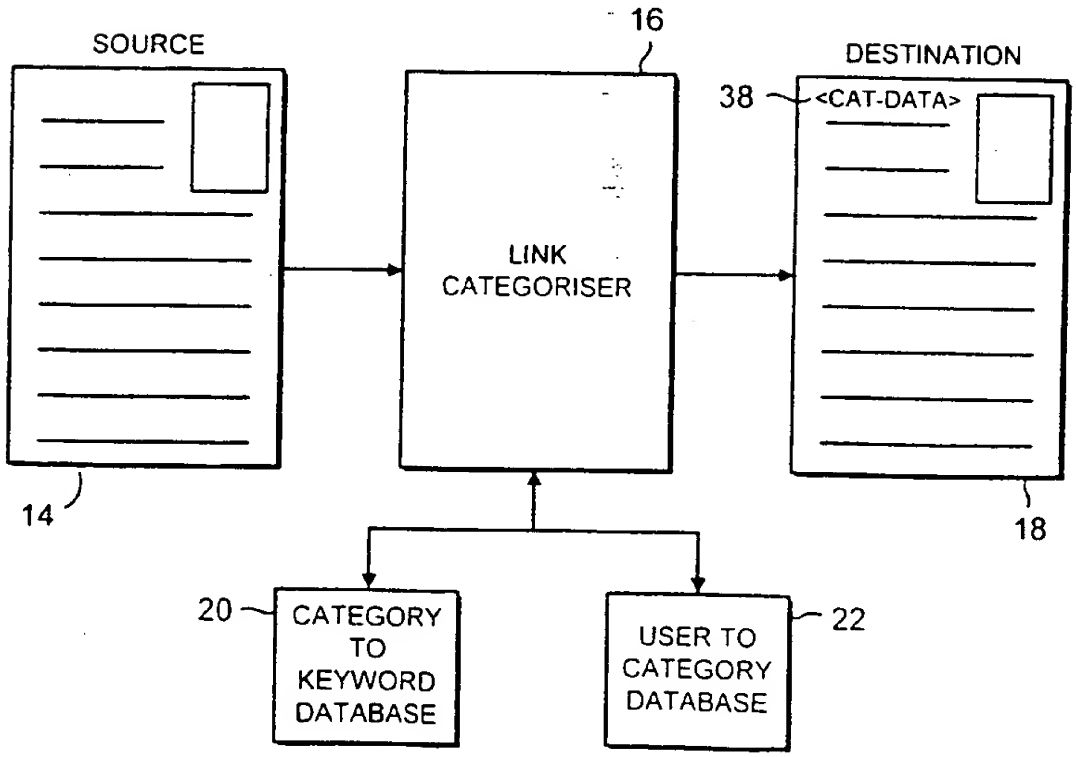

SOURCE
16
LINK CATEGORISER
DESTINATION
38
<CAT-DATA>
14
18
20
CATEGORY TO KEYWORD DATABASE
USER TO CATEGORY DATABASE
22

FIG. 2

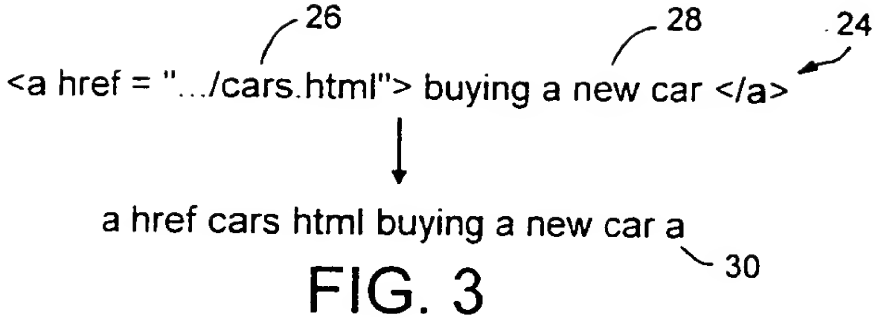


FIG. 3

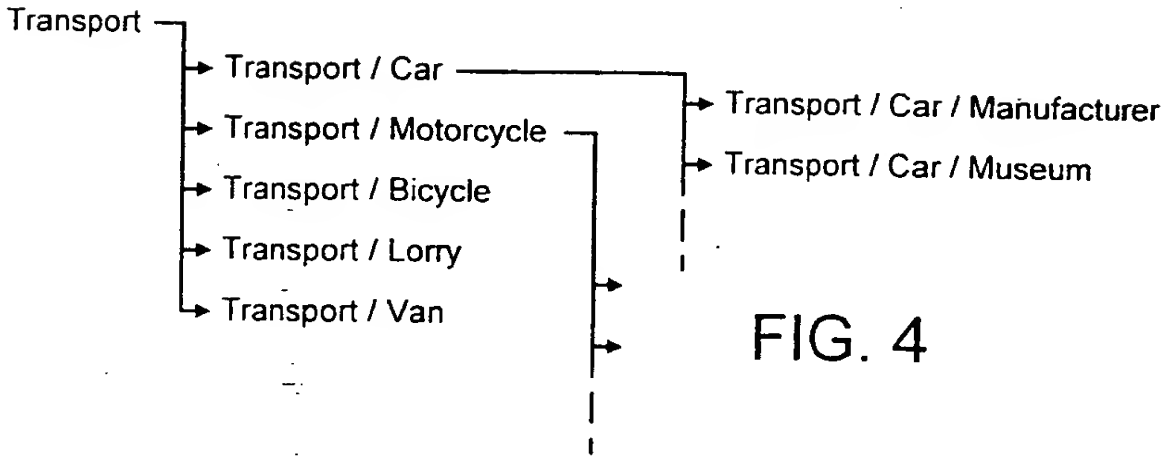


FIG. 4

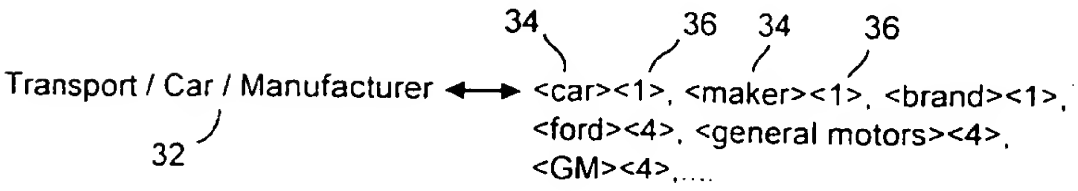


FIG. 5

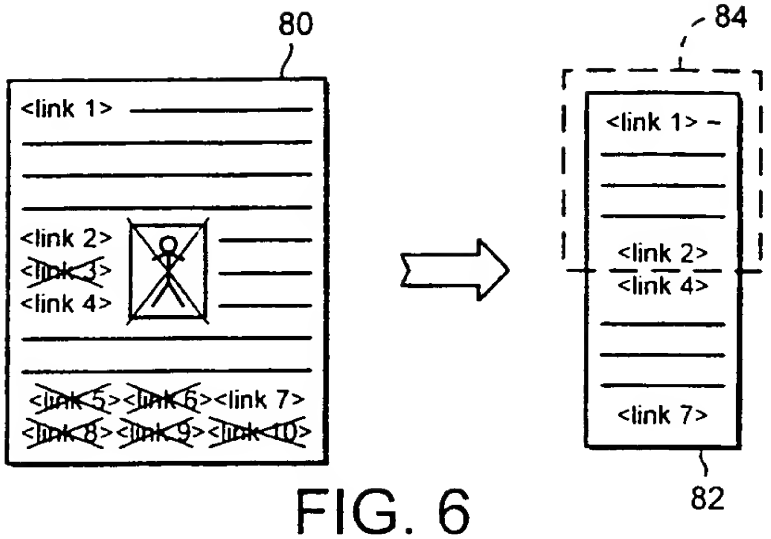

80
<link 1>
<link 2>
<link 3>
<link 4>
<link 5><link 6><link 7>
<link 8><link 9><link 10>
84
<link 1>
<link 2>
<link 4>
<link 7>
82

FIG. 6

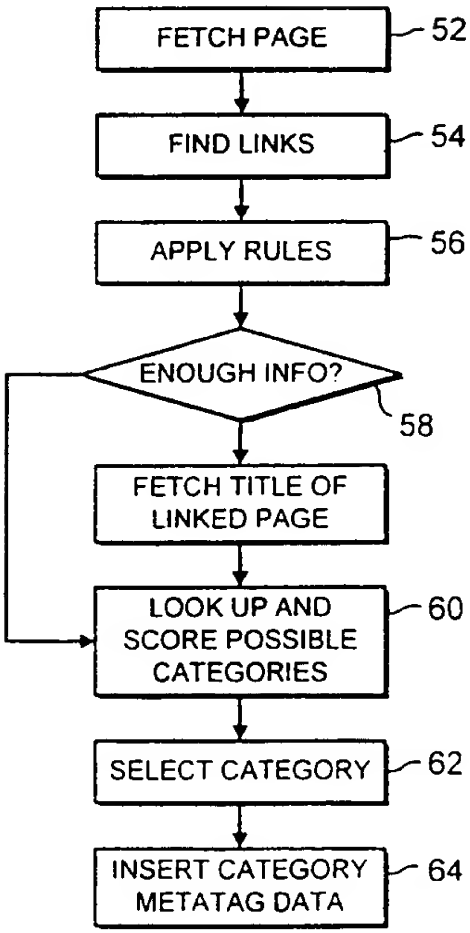

FETCH PAGE
52
FIND LINKS
54
APPLY RULES
56
ENOUGH INFO?
58
FETCH TITLE OF LINKED PAGE
LOOK UP AND SCORE POSSIBLE CATEGORIES
60
SELECT CATEGORY
62
INSERT CATEGORY METATAG DATA
64

FIG. 7

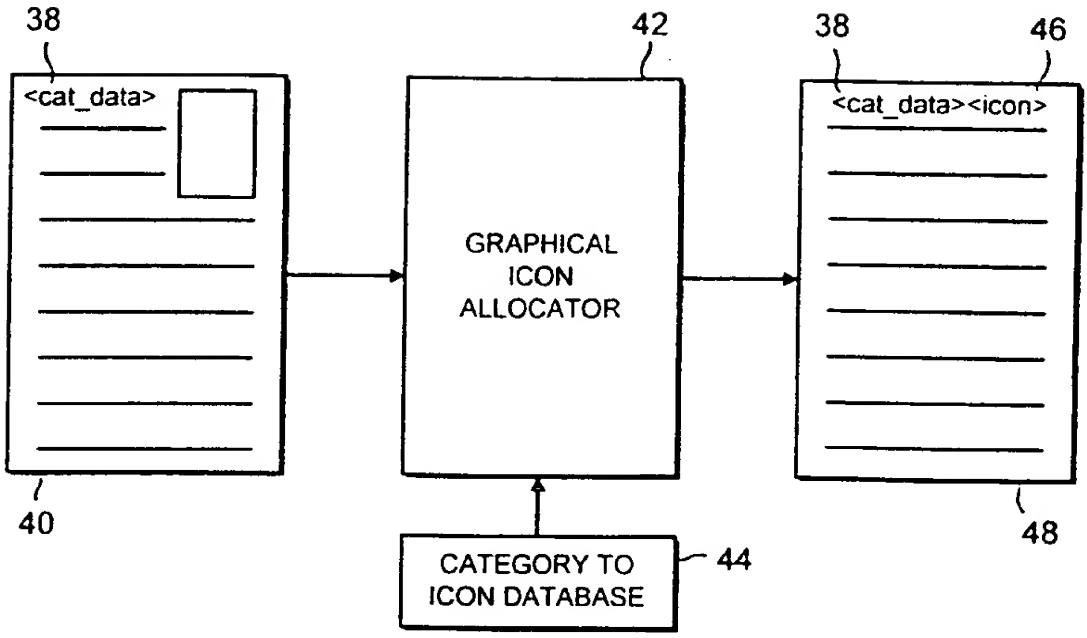

38
<cat_data>
40
42
GRAPHICAL ICON ALLOCATOR
CATEGORY TO ICON DATABASE
44
38
46
<cat_data><icon>
48

FIG. 8

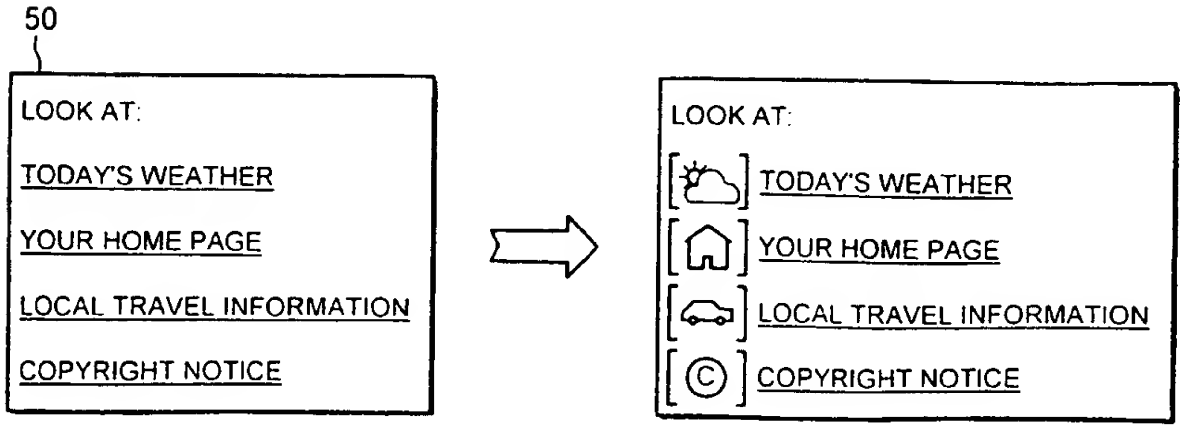

50
LOOK AT:
TODAY'S WEATHER
YOUR HOME PAGE
LOCAL TRAVEL INFORMATION
COPYRIGHT NOTICE
LOOK AT:
TODAY'S WEATHER
YOUR HOME PAGE
LOCAL TRAVEL INFORMATION
COPYRIGHT NOTICE

FIG. 9

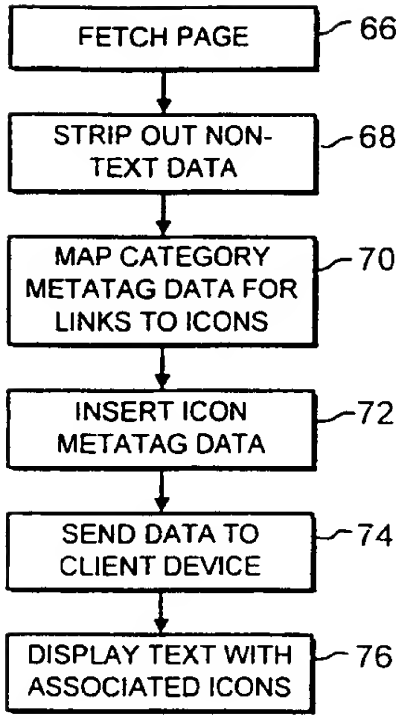

FETCH PAGE
66
STRIP OUT NON-TEXT DATA
68
MAP CATEGORY METATAG DATA FOR LINKS TO ICONS
70
INSERT ICON METATAG DATA
72
SEND DATA TO CLIENT DEVICE
74
DISPLAY TEXT WITH ASSOCIATED ICONS
76

FIG. 10

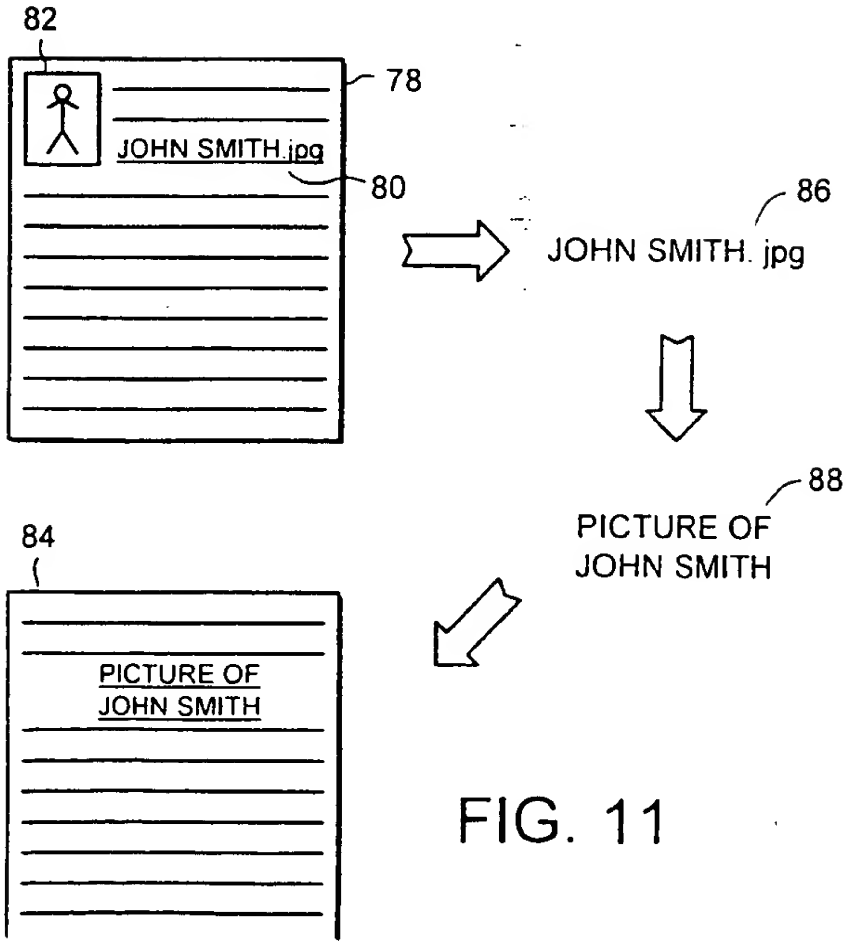

82
78
JOHN SMITH.jpg
80
86
JOHN SMITH. jpg
88
PICTURE OF JOHN SMITH
84
PICTURE OF JOHN SMITH

FIG. 11

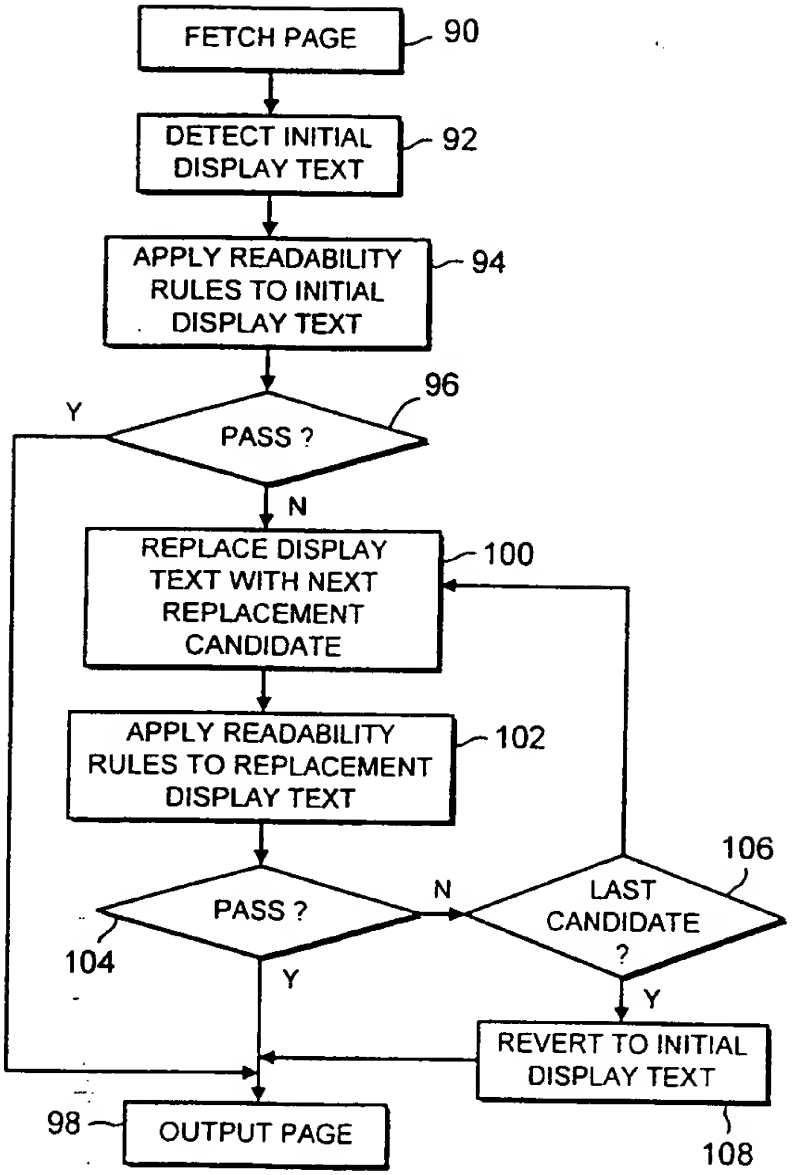


FIG. 12

A	"file_001234.doc"	→	"contact details page"
B	"link 1"	→	"transport - cars"
C	"please select this link to see a full list of our products available in your country"	→	"select products country"
D	"products.doc"	→	"products"

FIG. 13

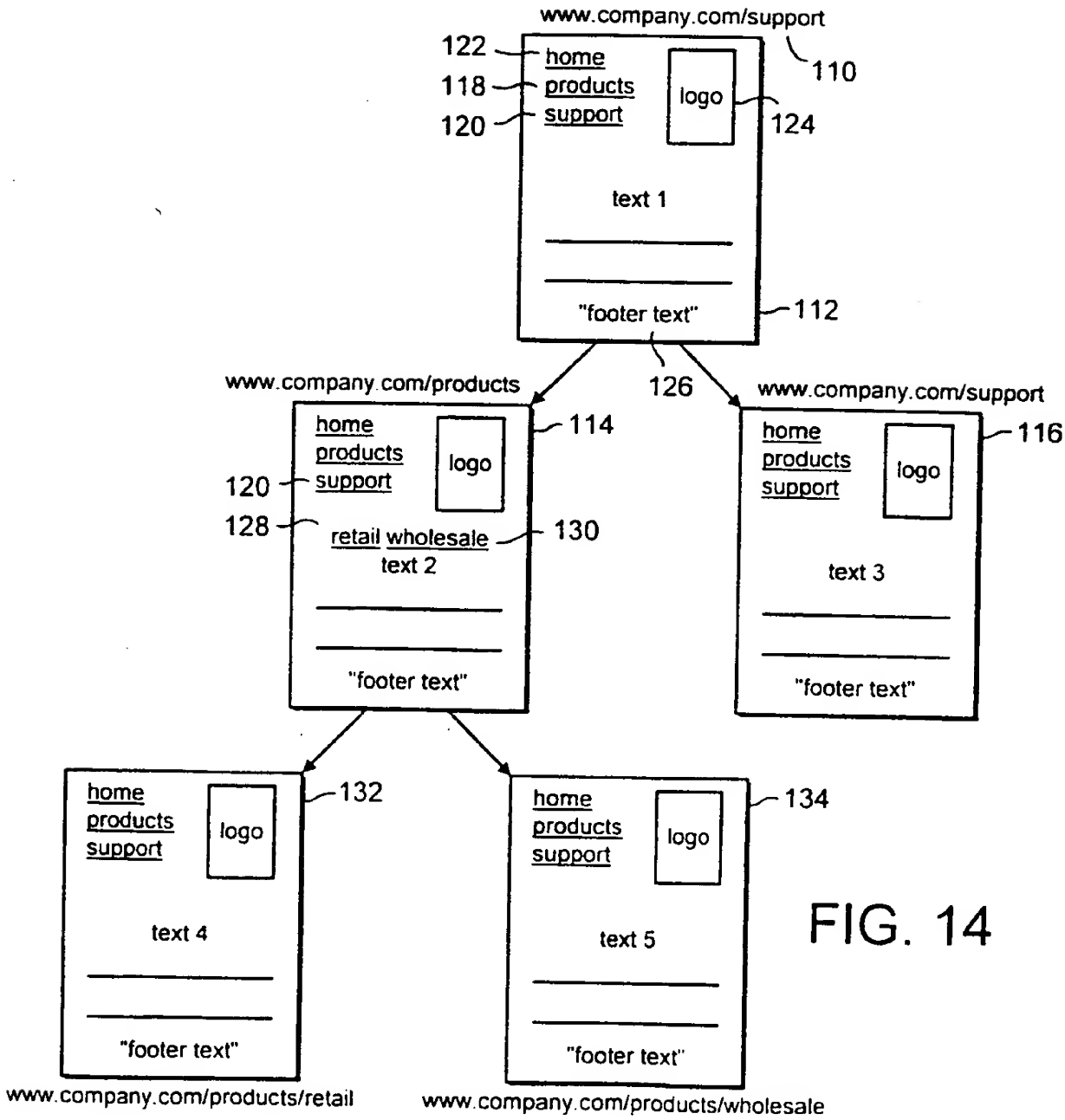

www.company.com/support
122
home
118
products
120
support
logo
110
124
text 1
"footer text"
112
126
www.company.com/products
114
home
products
support
logo
120
128
retail wholesale
text 2
130
"footer text"
www.company.com/support
116
home
products
support
logo
text 3
"footer text"
132
home
products
support
logo
text 4
"footer text"
www.company.com/products/retail
134
home
products
support
logo
text 5
"footer text"
www.company.com/products/wholesale

FIG. 14

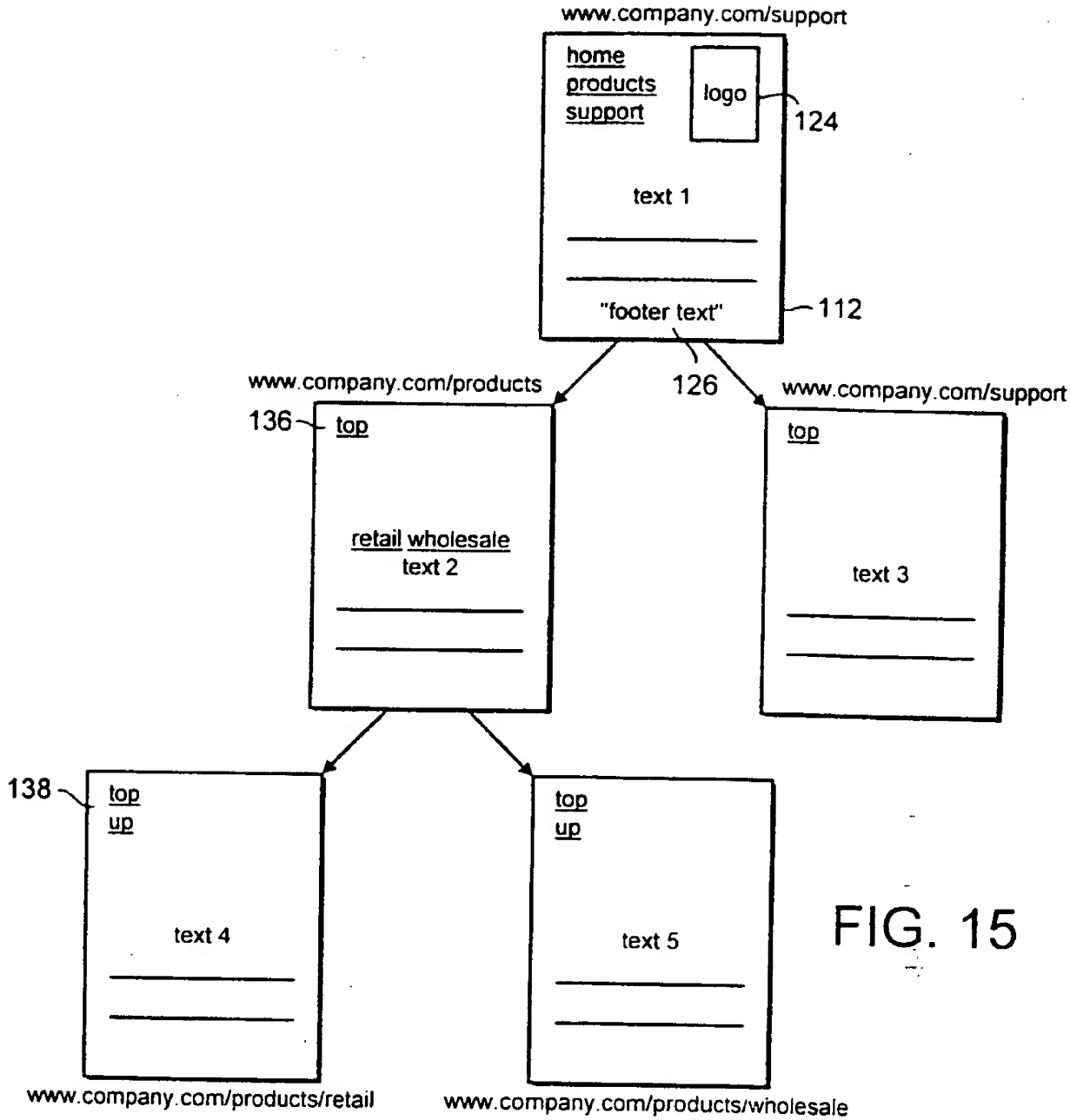
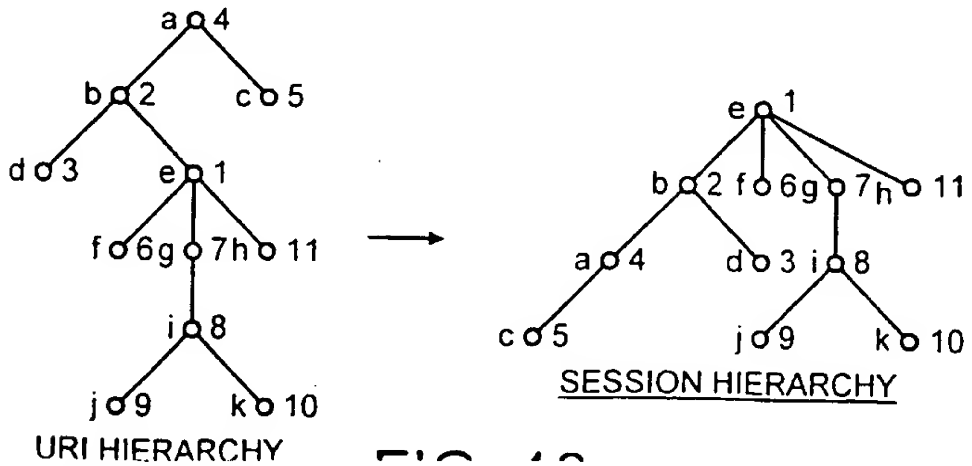

www.company.com/support
home
products
support
logo
124
text 1
"footer text"
112
www.company.com/products
136
top
retail wholesale
text 2
126
www.company.com/support
top
text 3
138
top
up
text 4
www.company.com/products/retail
top
up
text 5
www.company.com/products/wholesale

FIG. 15


a 4
b 2
c 5
d 3
e 1
f 6g 7h 11
i 8
j 9
k 10
URI HIERARCHY
e 1
b 2 f 6g 7h 11
a 4
d 3 i 8
c 5
j 9
k 10
SESSION HIERARCHY

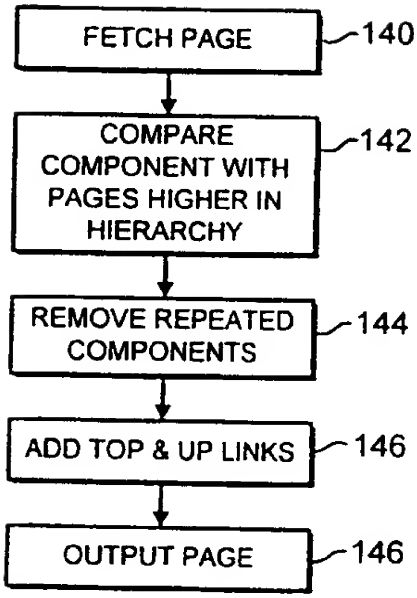

FETCH PAGE
140
COMPARE COMPONENT WITH PAGES HIGHER IN HIERARCHY
142
REMOVE REPEATED COMPONENTS
144
ADD TOP & UP LINKS
146
OUTPUT PAGE
146

FIG. 17

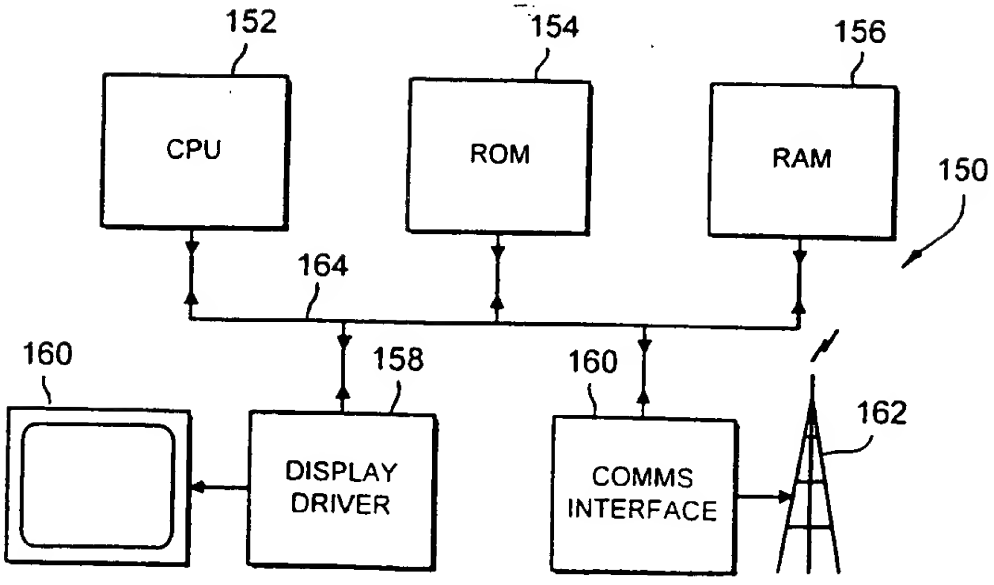

152
154
156
CPU
ROM
RAM
150
164
160
158
160
162
DISPLAY DRIVER
COMMS INTERFACE

FIG. 18